Notice of Allowability	Application No.	Applicant(s)	
	09/556,069	ALLAVARPU ET AL.	
	Examiner	Art Unit	
	Wen-Tai Lin	2154	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *appeal brief filed 6/29/2004 and telephone interview held on 2/11/2005*.

2. ☒ The allowed claim(s) is/are *1-34*.

3. ☒ The drawings filed on *4/21/2000* are accepted by the Examiner.

4. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) ☐ All b) ☐ Some* c) ☐ None of the:
      1. ☐ Certified copies of the priority documents have been received.
      2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
      3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

5. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.

6. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
      1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____.
   (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

7. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☒ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. ☐ Information Disclosure Statements (PTO-1449 or PTO/SB/08), Paper No./Mail Date _____
4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. ☐ Notice of Informal Patent Application (PTO-152)
6. ☒ Interview Summary (PTO-413), Paper No./Mail Date _____ .
7. ☒ Examiner's Amendment/Comment
8. ☒ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____.

REASONS FOR ALLOWANCE

AND

EXAMINER'S AMENDMENT

1. The following is an examiner's statement of reasons for allowance:

The prior art of record does not teach or suggest individually or in combination a method for converting between data types expressed in abstract syntax notation and data types expressed in an interface definition language, wherein the abstract syntax notation comprises a language for describing data and the interface definition language comprises a language for implementing interfaces to managed objects, wherein a same converter interface is accessed for a plurality of communications pertaining to each of the managed objects, and wherein the plurality of communications comprise differently typed data.

2. An examiner's Amendment to the record appears below. Should the changes and/or additions be unacceptable, an amendment may be filed as provided by 37 C.F.R. 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the Issue Fee.

3. Authorization for the examiner's amendment was given in a telephone interview with Mr. Robert Kowert, reg. no.39255, on February 11, 2005.

4. In the claims, please replaces claims 1, 13, 25, 31 and 33 to the following:

1. (Currently amended) A method for mapping managed object metadata, the method comprising:

receiving a plurality of communications each pertaining to a different one of a plurality of managed objects from a management server for the managed objects, wherein each communication comprises data typed according to an abstract syntax notation, wherein respective communications pertaining to each of two or more of the plurality of managed objects comprise differently typed data; and

accessing a converter interface for each communication for converting the abstract syntax notation data types of each communication to interface definition language data types, wherein the same converter interface is accessed for each of the managed objects such that said converting is generic to the managed objects, wherein said converting comprises:

inputting a first data type from a first set of data types, wherein the first set of data types is expressed in the abstract syntax notation, and wherein the abstract syntax notation comprises a language for describing data;

determining a corresponding second data type from a second set of data types, wherein the second set of data types is expressed in the interface definition language, wherein the interface definition language comprises a language for implementing interfaces to managed objects, wherein the interface definition language is operable across a plurality of platforms and across a plurality of programming languages, and wherein the interface definition language is class independent; and

returning the second data type.

13. (Currently amended) A method for mapping managed object metadata, the method comprising:

receiving a plurality of communications each pertaining to a different one of a plurality of managed objects from a manager for the managed objects, wherein each communication comprises data typed according to an interface definition language, wherein respective communications pertaining to each of two or more of the plurality of managed objects comprise differently typed data;

accessing a converter interface for each communication for converting the interface definition language data types of each communication to abstract syntax notation data types, wherein the same converter interface is accessed for each of the managed objects such that said converting is generic to the managed objects, wherein said converting comprises:

inputting a first data type from a first set of data types, wherein the first set of data types is expressed in the interface definition language, wherein the interface definition language comprises a language for implementing interfaces to managed objects, wherein the interface definition language is operable across a plurality of platforms and across a plurality of programming languages, and wherein the interface definition language is operable to provide a mapping which is applicable to any managed object class;

determining a corresponding second data type from a second set of data types, wherein the second set of data types is expressed in ~~an~~ the

abstract syntax notation, and wherein the abstract syntax notation comprises a language for describing data; and

returning the second data type.

25. (Currently amended) A computer system for mapping managed object metadata, wherein the system comprises:

a CPU;

a memory coupled to the CPU, wherein the memory stores program instructions executable by the CPU, and wherein the program instructions are executable to implement a generic converter for converting abstract syntax notation data types to interface definition language data types for metadata pertaining to a plurality of different managed objects, wherein respective metadata pertaining to two or more of the plurality of different managed objects comprises different abstract syntax notation data types, wherein the same converter interface is accessed for each of the managed objects, wherein the generic converter is configured to:

input a first data type from a first set of data types, wherein the first set of data types is expressed in the abstract syntax notation, and wherein the abstract syntax notation comprises a language for describing object data;

determine a corresponding second data type from a second set of data types, wherein the second set of data types is expressed in the interface definition language, wherein the interface definition language comprises a language for implementing interfaces to managed objects, wherein the interface definition language is operable across a plurality of platforms and across a plurality of

programming languages, and wherein the data types in the interface definition language are class independent; and

return the second data type.

31. (Currently amended) A carrier medium comprising program instructions for mapping managed object metadata, wherein the program instructions are computer-executable to implement:

receiving a plurality of communications each pertaining to a different one of a plurality of managed objects from a management server for the managed objects, wherein each communication comprises data typed according to an abstract syntax notation, wherein respective communications pertaining to each of two or more of the plurality of managed objects comprise differently typed data;

accessing a converter interface for each communication for converting the abstract syntax notation data types of each communication to interface definition language data types, wherein the same converter interface is accessed for each of the managed objects such that said converting is generic to the managed objects, wherein said converting comprises:

inputting a first data type from a first set of data types, wherein the first set of data types is expressed in the abstract syntax notation, and wherein the abstract syntax notation comprises a language for describing data;

determining a corresponding second data type from a second set of data types, wherein the second set of data types is expressed in the interface definition language, wherein the interface definition language comprises a language for implementing interfaces to

managed objects, wherein the interface definition language is operable across a plurality of platforms and across a plurality of programming languages, and wherein the interface definition language is class independent; and

returning the second data type.

33. (Currently amended) A carrier medium comprising program instructions for mapping managed object metadata, wherein the program instructions are computer-executable to implement:

receiving a plurality of communications each pertaining to a different one of a plurality of managed objects from a manager for the managed objects, wherein each communication comprises data typed according to an interface definition language, wherein respective communications pertaining to each of two or more of the plurality of managed objects each comprise differently typed data;

accessing a converter interface for each communication for converting the interface definition language data types of each communication to abstract syntax notation data types, wherein the same converter interface is accessed for each of the managed objects such that said converting is generic to the managed objects, wherein said converting comprises:

inputting a first data type from a first set of data types, wherein the first set of data types is expressed in the interface definition language, wherein the interface definition language comprises a language for implementing interfaces to managed objects, wherein the interface definition language is operable across a plurality of platforms and across a plurality of programming languages, and wherein the

interface definition language is operable to provide a mapping which is applicable to any managed object class;

determining a corresponding second data type from a second set of data types, wherein the second set of data types is expressed in the abstract syntax notation, and wherein the abstract syntax notation comprises a language for describing data; and

returning the second data type.

5. The abstract is amended to within 150 words in length. See the attached page.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Wen-Tai Lin whose telephone number is (571)272-3969. The examiner can normally be reached on Monday-Friday (8:00-5:00) .

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Follansbee can be reached on (571)272-3964. The fax phone numbers for the organization where this application or proceeding is assigned are as follows:

(703)872-9306 for official communications; and

(571)273-3969 for status inquires draft communication.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Wen-Tai Lin

February 11, 2005

Wen-Tai Lin
2/11/05

ABSTRACT OF THE DISCLOSURE

A system and method for generic and dynamic mapping of managed object metadata. The data to be mapped may include type information about an attribute, action, or notification of a managed object. The first data type is entered into the mapping system, and then a corresponding second data type is determined and returned. In this manner, data types related to the attributes, events, or other parameters of managed objects are converted between various data description languages, such as OMG IDL and ASN1. A single interface is defined to describe substantially all data types and all managed objects. Having a single object interface for all TMN managed objects saves significant resources in that there is no need to statically compile and store separate interfaces for the many objects, which may increase the scalability of a network management system dramatically.